

Scheduling Algorithms for Wireless Ad-Hoc Sensor Networks

Cédric Florens, Robert McEliece

Department of Electrical Engineering
California Institute of Technology
Pasadena, California, USA
E-mail: {florens, rjm}@systems.caltech.edu

Abstract—In this paper, we define a simple discrete mathematical model for wireless ad-hoc sensor networks and study the problems of data distribution and data collection which arise in those networks. We show how those tasks can be optimally performed on tree sensor networks that use directional antennas. Furthermore we compare the performance of a network equipped with directional antenna elements with one equipped with omnidirectional antenna elements and show the former outperforms the latter by 50% at most on a line network.

I. INTRODUCTION

The advancement of Very Large Scale Integration (VLSI) technology has contributed much to the development of micro sensor systems. Such systems can combine signal processing, data storage, wireless communication capabilities and energy sources on a single chip. Possibly distributed over a wide area, networks of such devices can autonomously perform various sensing tasks such as environmental (seismic, meteorological) monitoring and military surveillance [1]. These networks are referred to as wireless ad-hoc sensor networks or simply sensor networks. In sensor networks while each node may be mobile, it is typically the case that once the target site of their sensing application is reached a semi permanent stationary configuration is adopted for the purpose of gathering information.

In the area of general ad-hoc networks as well as sensor webs, research has focused on routing [2], medium access control (MAC) [3] [4] and physical layer [5]. [6] and [7] are protocol suites specifically designed for sensor webs. Theoretical results regarding capacity of general static ad-hoc networks first appeared in [8]. Also relevant to our research is the so called packet routing problem which consists in moving packets of data from one location to another as quickly as possible in a network and has been extensively studied in conjunction with wireline network models [9]. To the best of our knowledge no results specific to sensor networks had yet been derived.

We imagine a sensor network as having two main phases of operation (in steady state, after the nodes have organized themselves into a network). In a first phase or measuring phase, area monitoring results in an accumulation of data at each sensor, in the second phase or data transfer, the collected data is transmitted to some processing center located within the sensor network. In this paper we investigate the efficiency limits of such data transfers.

This work was performed at Caltech and supported by The Lee Center for Networking.

This paper is organized as follows: In section II we give a possible model for a sensor network. We considered two types of networks, one where nodes are equipped with directional antennas and one where nodes are equipped with omnidirectional antennas. We present our results for the former in section III and briefly present our results regarding the latter in section IV. We present a comparison analysis of the two systems in section V and conclude in section VI.

II. MODEL AND PROBLEM STATEMENT

We define a sensor network as a finite collection of n identical sensor nodes $\{N_1, \dots, N_n\}$. Each node N_i is associated with an integer p_i that represents the number of data packets stored at this node at the end of the measuring phase. There is one special node denoted N_0 , the processing center, which we will refer to as the base station (BS). All the nodes including the BS have a common transmission range r . A node (BS included) cannot receive and transmit at the same time. The interference model as defined in [8] is adopted here. That is, a transmission from node N_i to node N_j where $i, j \geq 0$ is successful if for every other node N_k , $k \geq 0$ simultaneously transmitting:

$$|N_i - N_j| \leq r, |N_k - N_j| \geq (1 + \delta)r, \delta > 0 \quad (1)$$

We assume in our model that time is slotted and a one hop transmission consumes one time slot (TS). The network is further assumed to be synchronous. A node can only transmit/receive one data packet per TS. Multiple transmissions may occur within the network in one TS under this interference model by virtue of spatial separation. Our network may be represented as a weighted rooted graph $\{V, E, \mathbf{p}\}$ where $V = \{N_0, \dots, N_n\}$, E denotes the set of links and $\mathbf{p} = (p_1, \dots, p_n)$. In this graph model the root represents the BS (N_0) and an edge represents an existing wireless connection (a link) between two sensor nodes, or a sensor node and the BS (a necessary condition for that connection to be present is that the distance between two nodes is less or equal than the transmission range r). By its nature this link is single duplex bidirectional. Our goal is to route the data contained at each node to the BS as efficiently as possible. We refer to this as the *data collection problem*.

III. DIRECTIONAL ANTENNA SYSTEMS

A. Line Networks

In this subsection we consider a line network (an example of which is given in Fig. 1) of sensor nodes. A BS is placed at

one end of the network. We assume sensor nodes are regularly placed along the network. We denote by d the distance between any 2 nodes. Assume each node is equipped with directional antennas allowing transmissions over a distance r where $d < r < 2d$. Further assume that δ is such that $(1+\delta)r < 2d$. It is straightforward to extend the following results to more general line networks where nodes are randomly placed along a line and to different values of r, δ , as long as end to end connectivity of the network is ensured. Let's denote node N_i by its distance to the BS in number of hops, that is, i . We denote $i \rightarrow i+1$ a transmission from node i to node $i+1$. Our goal is to determine the minimal duration of the transfer phase and an associated optimal communication strategy (Note that in general such a strategy is not unique).

For purpose of solving this problem we look initially at the following converse problem (which we shall subsequently refer to as the *distribution problem*); instead of nodes sending their respective packets to the BS, assume the BS is to transmit data packets to nodes. The data transfer efficiency remains our concern. This problem is of separate interest in sensor networks. We propose the following simple greedy algorithm for solving the distribution problem. We shall prove subsequently it is optimal. The BS is to send first data packets destined for the furthest node, then data packets for the second furthest one and so on, as fast as possible while respecting the channel reuse constraints. Nodes between the BS and its destinations are required to forward packets as soon as they arrive (that is in the TS following their arrival). Following is algorithm 1 running at the BS.

Given a line network (represented by the vector $Network = \mathbf{p}$), it dictates the BS actions at each time step: remain idle ($action = 0$) or transmit ($action = 1$). The result is stored in the vector $action$. When an action is chosen the right packet is to be handed over to the BS for transmission. One might assume that there is a stack of data packets correctly ordered with respect to the distance to the BS and that that stack is being updated after each BS action so that a packet is popped off the stack as it is transmitted.

Algorithm 1 Determines BS actions

input: Network
output: action
1: $step \leftarrow 1, legal \leftarrow 1, packets_left \leftarrow \sum_i Network(i)$
2: **while** $packets_left \neq 0$ **do**
3: **if** $legal$ **then**
4: $action(step) \leftarrow 1$
5: $packets_left \leftarrow packets_left - 1$
6: $legal \leftarrow 0$
7: **else**
8: $action(step) \leftarrow 0$
9: $legal \leftarrow 1$
10: **end if**
11: **if** $packets_left < Network(1)$ **then**
12: $legal \leftarrow 1$
13: **end if**
14: $step \leftarrow step + 1$
15: **end while**

The procedure is illustrated in the example of Fig. 1 where $V = \{0, 1, 2, 3, 4, 5, 6, 7\}$, $E = \{(i, i+1), 0 \leq i \leq 6\}$, $\mathbf{p} = (2, 0, 0, 0, 3, 0, 1)$, $d < r < 2d$, $(1+\delta)r < 2d$. The schedule of transmissions as determined by algorithm 1 is drawn be-

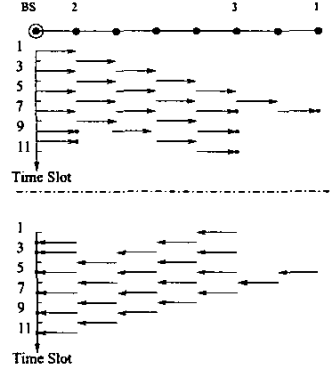


Fig. 1. 8-node line network ($p_1 = 2, p_2 = p_3 = p_4 = p_6 = 0, p_5 = 3, p_7 = 1$) followed by optimal transmission schedules for the distribution (upper schedule) and collection (lower schedule) problems. They are symmetric of one another. The job is performed in 11 TS

low the network for the distribution and collection problems respectively. Either way it is performed in 11 TS.

Next we determine the performance of our algorithm in general. Denote T_i the last busy TS at node i in the execution of our distribution algorithm (In the previous example, we have $T_1 = 10, T_2 = 9, T_3 = 10, T_4 = 11, T_5 = 11, T_6 = 7, T_7 = 7$). Clearly then our algorithm runs in $\max_{1 \leq i \leq n} \{T_i\}$. T_i is a function of the distance to the BS, the number of packets destined for node i as well as the number of packets forwarded by node i .

Lemma 1: Assuming $p_i = 0$ for $i > n$, we have:

$$T_i = \begin{cases} i - 1 + 2 \sum_{j \geq i+1} p_j & \text{if } p_i = 0 \\ p_1 + 2 \sum_{j \geq 2} p_j & \text{if } i = 1 \text{ and } p_1 \geq 1 \\ i - 2 + 2 \sum_{j \geq i} p_j & \text{if } i \geq 2 \text{ and } p_i \geq 1 \end{cases} \quad (2)$$

Proof:

$\forall i \geq 1$, node i is idle the first $i - 1$ TS. It forwards $\sum_{j \geq i+1} p_j$ data packets to further nodes and receives p_i data packets that are destined for itself. Forwarding a data packet consists in receiving that data packet and transmitting it right away and therefore a node involved in forwarding one data packet will remain busy two consecutive TS. Receiving a data packet on the other hand consumes only one TS but in our scheme forces node $i \geq 2$ to remain silent in the following TS. Therefore,

$$p_1 \geq 1 \Rightarrow T_1 = 2 \sum_{j \geq 2} p_j + p_1$$

$$p_i \geq 1, i > 1 \Rightarrow T_i = (i - 1) + 2 \sum_{j \geq i+1} p_j + 2(p_i - 1) + 1$$

$$p_i = 0 \Rightarrow T_i = (i - 1) + 2 \sum_{j \geq i+1} p_j$$

Clearly the maximum of T_i is obtained over the set $\{i \geq 1 \mid p_i \neq 0\}$. We define, for a given sensor network, $T_u(\mathbf{p})$ the minimum length of a time schedule over all time schedules that perform the distribution job. In particular we have:

Lemma 2:

$$T_u(\mathbf{p}) \leq \max_{\{i \geq 1 \mid p_i \neq 0\}} T_i \quad (3)$$

Next we derive a lower bound on $T_u(\mathbf{p})$.

Lemma 3: Assuming $p_i = 0$ for $i > n$, we have:

$$T_u(\mathbf{p}) \geq \max_{1 \leq i \leq n} (i - 1 + p_i + 2 \sum_{j \geq i+1} p_j) \quad (4)$$

Proof: Node i has to forward $\sum_{j \geq i+1} p_j$ data packets to further nodes. Forwarding one data packet consists in receiving and transmitting that data packet and therefore results in a two TS consumption (per forwarded packet). Besides it is itself the destination of p_i data packets. Each received data packet costs at least one TS. Furthermore node i can't be active before it receives a data packet which takes at least $i - 1$ TS. Therefore $S_i = 2 \sum_{j \geq i+1} p_j + p_i + (i - 1)$ is a lower bound on any time schedule for all i . Hence $\max_{1 \leq i \leq n} S_i$ is a lower bound on $T_u(\mathbf{p})$. ■

Finally we prove that the lower bound on $T_u(\mathbf{p})$ derived in lemma 3 equals the upperbound derived in lemma 2 and hence that the proposed schedule is optimal.

Theorem 4: Assuming $p_i = 0$ for $i > n$, we have:

$$T_u(\mathbf{p}) = \max_{1 \leq i \leq n} (i - 1 + p_i + 2 \sum_{j \geq i+1} p_j) \quad (5)$$

Proof: Clearly $S_1 = T_1$ and $S_i = T_i \forall i \geq 2$ if $p_i \leq 1$. If $p_i > 1$ then $T_i > S_i$ but either $p_{i-1} = 0$ then $S_{i-1} = T_i$ or $p_{i-1} \geq 1$ then $S_{i-1} > T_i$. ■

Corollary 5: In the particular case where no two consecutive components of vector \mathbf{p} equal zero, Theorem 4 reduces to:

$$T_u(\mathbf{p}) = p_1 + 2 \sum_{i \geq 2} p_i \quad (6)$$

We now return to the data collection problem. The construction of a schedule here is based on the symmetry of the operations of distribution and collection. A time schedule that is symmetric to the distribution problem's schedule with respect to a fictive horizontal axis (see example in Fig. 1) provides us with an optimal solution, the time to transmit data packets from nodes to the BS being indeed the same as the time to carry out the converse operation (and being therefore minimal). In particular a transmission $i \rightarrow i + 1$ occurring at TS j in the distribution problem is a transmission $i + 1 \rightarrow i$ occurring at TS $T_u(\mathbf{p}) + 1 - j$ in the collection problem. Since the solution to one problem gives us the solution to the other, we only consider the distribution problem in the sequel. Note that an additional issue is raised in the data collection case; indeed the described algorithms don't require the network to be synchronous in the distribution case (so the algorithms may be run in a distributed way) whereas they do in the data collection case.

B. 2-branch networks

Consider now a line network and place the BS anywhere on that line. Another way to look at this problem is to consider it as a two branch line network (the two branches being respectively represented by the data vectors \mathbf{p} and \mathbf{q}) where a one branch line network is the previously considered case. We denote $T_u(\mathbf{p}, \mathbf{q})$ the optimal performance achievable on a 2-branch network. A general algorithm to distribute data on multibranch networks is proposed in the next section. Our procedure is illustrated in the following example (Fig. 2) in the case where only two branches are present. It is subsequently

proved to be optimal. The BS transmits in the direction that requires most work (as determined by equation (5)) first and then alternates transmissions between the two branches. Furthest nodes are served first then second furthest and so on. A data packet is automatically forwarded in the slot that followed its reception by a non destination node.

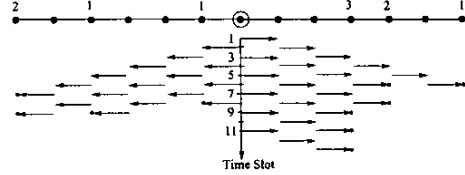


Fig. 2. Optimal distribution schedule in a 2-branch sensor network. The job is performed in 13 TS

Theorem 6: Without loss of generality we may assume $T_u(\mathbf{p}) \geq T_u(\mathbf{q})$, then:

$$T_u(\mathbf{p}, \mathbf{q}) = \begin{cases} \max(T_u(\mathbf{p}) + 1, \sum_{i \geq 1} p_i + q_i) & \text{if } T_u(\mathbf{p}) = T_u(\mathbf{q}) \\ \max(T_u(\mathbf{p}), \sum_{i \geq 1} p_i + q_i) & \text{if } T_u(\mathbf{p}) > T_u(\mathbf{q}) \end{cases} \quad (7)$$

Proof:

$$T_u(\mathbf{p}) = T_u(\mathbf{q}) \Rightarrow T_u(\mathbf{p}, \mathbf{q}) \geq \max(T_u(\mathbf{p}) + 1, \sum_{i \geq 1} p_i + q_i)$$

$$T_u(\mathbf{p}) > T_u(\mathbf{q}) \Rightarrow T_u(\mathbf{p}, \mathbf{q}) \geq \max(T_u(\mathbf{p}), \sum_{i \geq 1} p_i + q_i)$$

It is easy to see why the above described algorithm achieves this lower bound (consider for example the case $T_u(\mathbf{p}) = T_u(\mathbf{q})$). Indeed either the algorithm takes $T(\mathbf{p}) + 1$ TS to perform the job or it takes T'_{-1} (resp. T'_1) defined as the last busy TS at distance 1 to the left (resp. to the right) from the BS. If it so T'_{-1} (resp. T'_1) equals $\sum_{i \geq 1} p_i + q_i$. ■

C. Multibranch networks

The algorithm running at the BS determines at each TS toward which branch transmit, if transmission is possible at all. The direction of transmission is greedily decided, based on estimates (one per branch) of the completion time of the data transfer. Initial estimate for a given branch is determined by equation (5). The legal direction associated with the biggest estimate is chosen (a legal transmission is one that respects the channel reuse constraints, so for example it is not legal for our algorithm to transmit in two successive TS toward a given node located at distance greater than 2 from the BS), ties being broken randomly. When no legal direction exists the BS remains idle. After a decision has been made (transmit toward a particular direction or stay idle) the estimates at each branch are updated according to the following rule: if a legal direction was not chosen, its new estimate is its old estimate plus one. Illegal direction estimates remain unchanged. The idea is to minimize at each TS the overall estimate of the transmission time.

Next we illustrate the procedure on an example (Fig. 3). In the

accompanying table, we list data transfer completion time estimates at each TS and the corresponding decision made by the BS. As previously stated the initial completion time estimates are computed using equation (5). The table reads as follows. TS 1: All 4 transmission directions are legal. The BS chooses to transmit toward branch A. At TS 2, transmitting toward A is not a legal move, the legal transmission direction associated with the biggest estimate is B, etc. Along a given branch, the packets destined for furthest nodes are sent first by the BS. As for the other nodes they merely forward the data packets of which they are not recipients (a packet is transmitted in the following TS that it was received). In this example the algorithm performs in 10 TS (an obvious lower bound is 8 TS corresponding to 8 data packets). It turns out 10 TS is indeed a lower bound on the execution time of any schedule. Here we omit the general case proof. It may be found in [10].

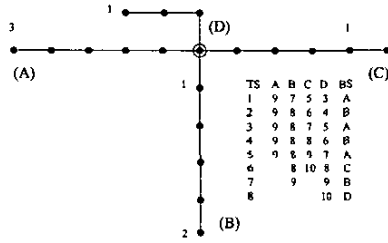


Fig. 3. Optimal distribution schedule for BS on a 4-branch sensor network. The completion time is 10 TS.

D. Tree networks, case where degree of base station is 1

Throughout this paragraph we assume that the degree of the root of the considered graphs is one.

Definition 7: We define the *equivalent linear network* (G_l, E_l, \mathbf{p}_l) of a network (G, E, \mathbf{p}) . If $G = \{N_0, N_1, \dots, N_n\}$ and $\mathbf{p} = (p_1, \dots, p_n)$ then $G_l = \{0, 1, \dots, m\}$, $E_l = \{(i-1, i), 1 \leq i \leq m\}$ and $\mathbf{p}_l = (p_{l1}, \dots, p_{lm})$ where $m = \max(d(N_0, N_i))$ and $p_{lj} = \sum_{i: d(N_0, N_i)=j} p_i$. We illustrate a tree network in Fig. 4 ($n = 14, m = 7$); its equivalent linear network is shown in Fig. 1.

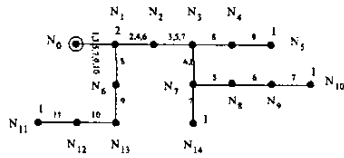


Fig. 4. A 15-node tree network with degree of BS=1, the equivalent linear network is drawn in Fig. 1. Transmission TS are written next to the edges.

The equivalent linear network's schedule may serve as a schedule for the initial tree network. Next we explain how transmission time slots for (G_l, E_l, \mathbf{p}_l) (determined by running algorithm 1) may be mapped onto (G, E, \mathbf{p}) . Consider an element in E , say (N_{i_0}, N_{j_0}) , such that $d(N_0, N_{i_0}) = \alpha$ (hops). Based on the number of data packets N_{j_0} has to forward, say f_{j_0} , we shall allocate transmission time slots to edge (N_{i_0}, N_{j_0}) . Define $E_\alpha = \{(N_i, N_j) \in$

$E | d(N_0, N_i) = \alpha\}$. Each packet P follows a path $path(P)$ from the BS to its destination node where $path(P)$ denotes the finite sequence of edges (e_1, \dots, e_k) traversed in that order by P . For convenience we shall write $path(P)$ as the sequence of vertices $(vertices(e_1), \dots, vertices(e_k))$. We define $\mathfrak{P}_\alpha = \{P | \exists e \in E_\alpha \cap path(P)\}$. We define $\mathcal{T}_\alpha = \{\text{TS used by } (\alpha, \alpha+1) \in E_l\}$. We have: $|\mathfrak{P}_\alpha| = \sum_{(N_i, N_j) \in E_\alpha} (p_j + f_j) = \sum_{k > \alpha} p_{lk} = |\mathcal{T}_\alpha|$. Thus one may

define a one to one correspondence g between \mathfrak{P}_α and \mathcal{T}_α that associates the packet P with the longest path in \mathfrak{P}_α , with the TS with the smallest index in \mathcal{T}_α ; the packet P with second longest path, with the TS with second smallest index and so on. We finally define $\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})} = \{P | (N_{i_0}, N_{j_0}) \in path(P)\} \subseteq \mathfrak{P}_\alpha$. (N_{i_0}, N_{j_0}) is associated with time slots $g(\mathfrak{P}_\alpha^{(N_{i_0}, N_{j_0})})$. In the example of Fig. 4, we have: $\{P\} = \{P_1, P_2, \dots, P_6\}$ where the first packet is characterized by $path(P_1) = (N_0, N_1, N_2, N_3, N_7, N_8, N_9, N_{10})$, the second one by $path(P_2) = (N_0, N_1, N_2, N_3, N_4, N_5)$, the third one by $path(P_3) = (N_0, N_1, N_6, N_{13}, N_{12}, N_{11})$, the fourth one by $path(P_4) = (N_0, N_1, N_2, N_3, N_7, N_{14})$, and finally the fifth and sixth ones by $path(P_5) = path(P_6) = (N_0, N_1)$. We also have $E_1 = \{(N_1, N_2), (N_1, N_6)\}$, $\mathfrak{P}_1 = \{P_1, P_2, P_3, P_4\}$, $\mathcal{T}_1 = \{2, 4, 6, 8\}$, and $\mathfrak{P}_1^{(N_1, N_2)} = \{P_1, P_2, P_4\}$. Thus edge (N_1, N_2) is associated with time slots $g(\mathfrak{P}_1^{(N_1, N_2)}) = \{2, 4, 6\}$. Thus algorithm 1 run on the equivalent linear network provides a BS transmission schedule. Intermediate nodes simply forward data packets to further nodes as they arrive (in the TS following their arrival). This requires a routing table at junction nodes.

Although an equivalent linear network has a reduced set of possible concurrent transmissions, this procedure produces an optimal transmission schedule. The proof omitted here because of space limitations is based on the fact that transmissions that can occur in one case and not in the other, that is 1) multiple transmissions from nodes at distance i (from BS) to nodes at distance $i+1$ and 2) simultaneous transmissions from node(s) at distance i to node(s) at distance $i+1$ on one hand and from node(s) at distance $i-1$ to node(s) at distance i on the other hand are not helpful in routing data faster. This is due to the fact that any path from the BS to a node necessarily includes link $(0, 1)$ which constitutes a bottleneck.

E. Tree sensor networks, general case

The results in the previous sections suggest the following algorithm for dealing with general tree networks.

- 1) linearize the subtrees attached to the BS according to the procedure described in III.D
- 2) apply multibranch algorithm described in section III.C to the resulting multibranch system

By combining results obtained in III.A, C and D, the procedure is clearly optimal for general tree networks.

IV. OMNIDIRECTIONAL ANTENNA SYSTEMS

Line networks

Our results readily extend to omnidirectional antenna systems. We have, for a one branch line network, using

an algorithm similar to algorithm 1 and keeping the notations/assumptions introduced in section III:

$$T_1 = \begin{cases} 3 \sum_{j \geq 3} p_j - 1 & \text{if } p_1 = 0, p_2 = 0 \text{ and } \sum_{j \geq 3} p_j \geq 1 \\ p_1 + 2p_2 + 3 \sum_{j \geq 3} p_j & \text{otherwise} \end{cases} \quad (8)$$

$$T_2 = 2p_2 + 3 \sum_{j \geq 3} p_j \quad (9)$$

$\forall i \geq 3$

$$T_i = \begin{cases} i - 2 + 3 \sum_{j > i} p_j & \text{if } p_i = 0 \text{ and } \sum_{j > i} p_j \geq 1 \\ i + 3 \sum_{j > i} p_j & \text{if } p_i = 1 \\ i - 3 + 3 \sum_{j \geq i} p_j & \text{if } p_i \geq 2 \end{cases} \quad (10)$$

Theorem 8:

$$T_o(\mathbf{p}) = \max_{1 \leq i \leq n} \{i - 1 + p_i + 2p_{i+1} + 3 \sum_{j > i+1} p_j\} \quad (11)$$

Omnidirectional antennas generate more channel reuse constraints and as a result require longer time schedules. For example if the network of Fig. 1 was equipped with omnidirectional antennas, the minimum schedule length would be 14 TS (this takes into account the fact that transmissions from node i to node $i + 1$ creates interferences at node $i - 1 \forall i \geq 1$). This is a 27% increase over the directional antenna system. Bounds on the relative performance of the two systems are derived in the next section.

V. COMPARISON BETWEEN OMNIDIRECTIONAL AND DIRECTIONAL SYSTEMS

In order to get a better intuition on how the two systems perform relative to one another, we give the following comparative result for a line network:

Theorem 9:

$$1 \leq \frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} < 1.5 \quad (12)$$

Proof: observe that:

- If the maximum of omnidirectional/directional T_i occurs in $j + 2$ then the maximum of omnidirectional S_i occurs in j whereas the maximum of directional S_i occurs in $j + 1$. Hence by equations (5) and (11), we get:

$$\Rightarrow \frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} = \frac{j - 1 + p_j + 2p_{j+1} + 3 \sum_{i \geq j+2} p_i}{j + p_{j+1} + 2 \sum_{i \geq j+2} p_i}$$

$$\text{Hence, } \frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} \geq 1.5 \Rightarrow 0.5j + 1 \leq p_j + 0.5p_{j+1} = 0$$

But this is not possible since $j \geq 0$.

- If the maximum of omnidirectional/directional T_i occurs in $j \in \{1, 2\}$ then the maximum of omnidirectional/directional S_i occurs in $j = 1$. Therefore by equations (5) and (11), we have:

$$\Rightarrow \frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} = \frac{p_1 + 2p_2 + 3 \sum_{j \geq 3} p_j}{p_1 + 2p_2 + 2 \sum_{j \geq 3} p_j}$$

$$\text{Hence, } \frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} \geq 1.5 \Rightarrow p_2 + 0.5p_1 \leq 0$$

Again this is impossible. Hence the right half of the inequality. ■

To see these bounds are tight, we look at the two following (typical) scenarios ($n \geq 3$):

- 1) $\mathbf{p} = (1, \dots, 1)$ By equations (6) and (11), $\frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} = \frac{3(n-1)}{2n-1} \xrightarrow{n \rightarrow \infty} 1.5$
- 2) $\mathbf{p} = (0, \dots, 0, p_n)$ By equations (5) and (11), $\frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} = \frac{n-3+3p_n}{n-2+2p_n}$ if $p_n > 1$; $\frac{T_o(\mathbf{p})}{T_u(\mathbf{p})} = 1$ if $p_n = 1$.

VI. CONCLUSIONS AND FUTURE WORK

We have proposed optimal strategies to distribute and collect data packets from a tree-like sensor network when the distributing/collecting node has degree 1 or 2. The exact performance times of such strategies have been derived. We have generalized those strategies to general tree networks. Finally we compared the performance of omnidirectional systems to directional ones. We are currently working on extending our comparison analysis between directional and omnidirectional systems. Furthermore we are looking at the impact of network cycles on the optimality of our algorithms.

ACKNOWLEDGMENTS

The authors would like to acknowledge the suggestions of Jeremy Thorpe.

REFERENCES

- [1] J. R. Agre, L. P. Clare, G. J. Potty, and N. P. Romanov, "Development platform for self-organizing wireless sensor networks," *Proceedings of SPIE, unattended ground sensor technologies and applications*, vol. 3713, pp. 257–268, 1999.
- [2] E.M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc wireless networks," *IEEE Personal Communications Magazine*, pp. 46–55, Apr. 1999.
- [3] V. Bharghavan, A. Demers, S. Shenkar, and L. Zhang, "MACAW: a media access protocol for wireless LAN," *Proceedings of the SIGCOMM 94 Conference of Communications architectures, protocols and applications*, p. 212–225, Aug. 1994.
- [4] J. H. Ju and V. O. K. Li, "TDMA scheduling design of multihop packet radio networks based on latin squares," *IEEE journal on selected areas in communications*, vol. 17, no. 8, pp. 1345–1352, Aug. 1999.
- [5] M. Pursley, "The role of spread spectrum in packet radio networks," *Proceedings of the IEEE*, Jan. 1987.
- [6] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, Oct. 2000.
- [7] Chalermek Intanagonwatt, Ramesh Govindan, and Deborah Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, Aug. 2000.
- [8] P. Gupta and P.R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, Mar. 2000.
- [9] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in $O(\text{congestion} + \text{dilation})$ steps," *Combinatorica*, vol. 14, pp. 167–186, 1994.
- [10] C. Florens and M. Franceschetti, "Scheduling, technical report,"